EEIRI: Efficient Encrypted Image Retrieval in IoT-Cloud

Zaid Ameen Abduljabbar[1], Ayad Ibrahim[1], Mohammed Abdulridha Hussain[1], Zaid Alaa Hussien[2], Mustafa A. Al Sibahee[3, 4], and Songfeng Lu [3, 4*]
[1]College of Education for Pure Sciences, University of Basrah, Basrah, Iraq
[e-mail: alsulamizaid@gmail.com]
[2]Southern Technical University, Basrah, Iraq.
[e-mail: zaid.alaa@stu.edu.iq]
[3]Shenzhen Huazhong University of Science and Technology Research Institute, Shenzhen 518063, China
[4]School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, 430074, China
[e-mail: lusongfeng@hust.edu.cn]
*Corresponding author: Songfeng Lu

Received October 1, 2017; revised November 20, 2018; revised January 21, 2019; revised February 28, 2019; accepted April 17, 2019; published November 30, 2019

Abstract

One of the best means to safeguard the confidentiality, security, and privacy of an image within the IoT-Cloud is through encryption. However, looking through encrypted data is a difficult process. Several techniques for searching encrypted data have been devised, but certain security solutions may not be used in IoT-Cloud because such solutions are not lightweight. We propose a lightweight scheme that can perform a content-based search of encrypted images, namely EEIRI. In this scheme, the images are represented using local features. We develop and validate a secure scheme for measuring the Euclidean distance between two descriptor sets. To improve the search efficiency, we employ the *k*-means clustering technique to construct a searchable tree-based index. Our index construction process ensures the privacy of the stored data and search requests. When compared with more familiar techniques of searching images over plaintexts, EEIRI is considered to be more efficient, demonstrating a higher search cost of 7% and a decrease in search accuracy of 1.7%. Numerous empirical investigations are carried out in relation to real image collections so as to evidence our work.

Keywords: Searchable encryption, secure image retrieval, IoT-Cloud, k-means clustering, SURF local feature, similarity measure

1. Introduction

The widespread utilization of more smart mobile devices to explore image inside extensive datasets is now more significant than ever. This importance is evident in numerous fields including that of social media [1] [2] business community [3], health-care [4], and in the recognition of criminals. Especially, on account of third and fourth generation lightweiht smart devices, clients can quickly associate with the Web or Internet and after that effectively hand-off, send, or obtain images while at the same time gathering various images from different sources. In any case, the administration and maintenance of these images makes noteworthy requests as far as storage and computational cost. These capabilities might be inaccessible for all smart cellphone clients, especially the individuals who are utilizing lightweight smart cellphone, for example, iPhones [5].

Considering the considerable progress of Cloud computing technology, various firms for example, Amazon, Google, and Microsoft have taken advantage of its technological benefits in order to offer its smart device consumers with more economical services including a larger storage capacity and data managing facilities on a greater scale [6]. Nevertheless, although the technological as well as the financial advantages of Cloud are numerous, the flow of images that are more sensitive in nature to untrusted *Cloud serves* (CSs), may serve to impinge on the privacy of certain images.

To combat unsolicited attempts to access such images, users generally choose to protect these images by ecrypting process before uploading them to the Cloud server. In any case, this procedure makes a boundary when utilizing customary encryption methods and while searching through images that have been encrypted. Additionally, in spite of the power processing of lightweight smart devices expanding, it is at present unfit to accomplish a similar limit of PCs.

In recent years, several of *searchable encryption* (SE) methodologies have been developed [7] [8] [9] [10] [11] [12] [13] [14] [15] that facilitate selective retrieval of encrypted documents by means of keyword searches. Within these systems a user would send a secured keyword to look for specific encrypted text documents. Other modern informational retrieval (IR) systems, such as Google, have introduced an original technology that allows clients to send an image as a query, search through the images that are retained on the database in question, and identify those images with comparable visual content. Original technological processes are referred as *content-based image retrieval* (CBIR). Therefore, a searchable encryption scheme that can proficiently and precisely manage image-based searches needs to be developed.

Lu et al. [16], highlighted the search of secure images via encrypted datasets as well as the preservation of data confidentiality using Min-Hash and *order preserving encryption* (OPE). However, on the downside it has 20% less search accuracy in comparison to systems which use Fisher vector as its basis [17] [18].

Afterward, a privacy- protection image search method that utilizes homomorphic encryption presented by Hsu et al. [19]. In spite of perceiving images with high precision, this plan is two to four requests of-size more costly, and requested the standard inclusion of the client all through the searching procedure. To the best our insight, a lightweight and secure plan is yet to be created for looking images over encrypted huge scale datasets through lightweight smart devices with insignificant misfortunes in inquiry time and accuracy when contrasted and other prevalent images scan techniques for plain texts, for example, [17] [18].

In essence, certain instances necessitate the need to have identical procedures in relation to IoT-Cloud such as when the confidentiality of images needs to be safeguarded. The following

example delineates the significance of securing image search inside IoT-Cloud. Envision a security organization has uploaded a substantial size database of touchy images identified with a potential fear based terrorist suspect to CSs as per its policy of completely using the advantages of Cloud innovation.

Those employed within security agencies may be required to verify the presence on the database of Cloud servers of any images relating to these suspects. Agents may utilize smart mobile devices in such cases in order to maintain a consistent pursuit of identifying suspects. However, for security purposes, neither the agents nor the agency want to reveal their stored sensative images and queries, respectively, except if a need access by authorized. This type of need can be determined by identifying any similarities between the image-based query of the agency and the Cloud database. Upon identifying a matching need, only images that are similar to the query can be sent by CS. Ideally, the agents and security agency should not reveal the database as well as the query image during this task, and the agents should only be informed of any similar images identified.

When image matching is used to identify an image linked to a terrorist suspect, the achievement of the security organization or agencey is rely upon the proficiency and accuracy with which they locate a matching image. The efficiency and accuracy of this process has significant implications for real life. From this perspective, commonly used cryptographic techniques, including homomorphic encryption, are not appropriate on account of their cost-intensiveness relative to the operating costs on plaintexts As a result, a strong and efficient system must be built to secure and process large quantities of encrypted images within IoT-Cloud using smart devices. Such process, which is as proficient and viable as searching for images on plaintext, can be referred to as proficient retrieval of encrypted image in IoT-Cloud (EEIRI).

The objectives of this paper are highlighted in the subsequent section. First, the manner in which we can look for and find similar images using smart devices technology and CS in a privacy-preserving manner and confidentiality; will be explored. Secondly, the duration and accuracy of a query on EEIRI are comparable with those of queries that are made using plaintext search methods. Thirdly, to accomplish high search precision and efficiency of EEIRI, we utilize local-feature *speeded up robust features* (SURF)-based CBIR with the appealing lightweight aspects through the use of Euclidean distance, as well-known metric to score matching images and employ matrix multiplications based encryption method to protect the confidentiality of image features, which makes the search cost is calculated on the encrypted images hold the same as doing over plaintexts. In order to enhance the efficiency of an image searching procedure, we have created a searchable index based on a tree, using the k-means clustering technique to classify an extensive images database into different clusters. Fourthly, our proposed plan tends to the issue of comparable image seeking to accomplish an adaptable inquiry. Fifthly, our proposed scheme can efficiently index a large-scale image databases, thereby reducing the storage requirements and runtime. Last, when considering the economics of communication, our system necessitates only a one round of communication between the query and CS interface, whereas others require several communications to take place. Overall, our plan denotes a significant advance towards pragmatic deployment of protection privacy information facilitated inside an IoT-Cloud server.

The following paper is divided into eight sections. Section 2 presents a review of existing scheme, while section 3 introduces a short explanation of the preliminaries .The problem statement and security definition in Section 4. Section 5 provides the proposed scheme design. EEIRI construction is then explained in Section 6. Additionally, Security analysis, outcomes of performance and investigations present in Section 7. Finally, Section 8 illustrates the

discoveries of this paper together in a synopsis and conclusion, and makes recommendations for further research.

2. Related Work

Searchable encryption methods based on text-based searches have proposed by several authors. The methods used to find particular encrypted data aims to retrieve encrypted content in plaintext so as to attain a decreased overhead in relation to economic cost. In spite of this, the majority of systems are associated with significant disadvantages. Bearing this in mind, we will highlight the related works associated with SEs as well as its disadvantages. Studies carried out in relation to CBIR-based SE will also be discussed.

Song et al. [7] first proposed a process that was based on a single-keyword searches. Despite achieving high precision, this technique does not facilitate similarity searching. To develop search proficiency and abstain from checking the whole text document to scan for a keyword as in [7] [8], the authors in [9] recommended SE procedures dependent on different index developments. In this manner, SE methods that utilization just a single keyword have been extended to utilize various and conjunctive keywords [10] [11] [12], thereby enhancing their precision and flexibility. Sadly, such plans might be not able decide and recoup composing mistakes inside the information, which are familiar in actuality applications. To address such trouble, a few SE methoda that use fuzzy keyword looks have been utilized to deal with any incorrect spellings that are faced with the search procedure [13] [14] [15].

In spite of the fact that those SE methods that utilization keywords include a few fascinating highlights, they are not appropriate for looking secure images over encrypted datasets. These plans are likewise constrained to surviving keyword sets, and keyword looking requires an exact textual description of accessible substance. On the other hand, a pre-characterized search scope for content-based image seek is at present unavailable.

Secure picture scans are performed inside the setting of scrambled sensetive image databases. The visual words extraction from images and the consequent development of indexes were recommended by Lu et al. [16], who used a progression of cryptographic methodologies, for example, OPE and randomized hash functions to give a more noteworthy insurance of privacy and upgrade the limit of rank-requested ability. In reality, OPE may release a lot of information to an untrusted server, in this way prompting fruitful attacks. Likewise, such a plan is just relevant to BOF image scans, and this has second rate seek accuracy up to 20% when diverged from the image look schemes dependent on Fisher vector [17] [18].

Although the work of Lu et al. [20] who pioneered a different protected search technique via the use of homomorphic encryption as well as bag-of-features (BOF) enabled a greater deal of confidentiality, this method decreased the precision and efficiency of searching when compared with [16] (refer analysis in [20]). In any case, experimental test outcomes demonstrate that the plan proposed in this paper, does not release any information to untrusted servers and achieves comparable search cost and precision when compared with engines for searching images over plaintexts [17] [18].

Perronnin et al. [17] utilized the Fisher vector inside the image search engine in order to precisely improve the accuracy of its searches. When compared with the BOF system it is thought that the Fischer vector is able to accomplish far more superior search results. Images that were searched using the Fisher vector as its basis improved the efficiency and precision in

various instances. Following [17], image search methods based on developed Fisher vector are proposed to more improve search accuracy in various instances [18].

Both [17] and [18] used the SIFT-based Fisher vector in the process of extracting local feature vectors [21] [22]. By contrast, we use the SURF method [23] [24], which produces fewer local features and outperforms the SIFT method in term of feature extraction processing speed. Subsection 3.1 explores in detail the differences between the local feature of SURF and SIFT.

In our paper, to look at the closeness of two images, we normalize their comparing SURF feature vectors. Afterward, dimensions of both vectors (and) can be reduced with a minimal loss has no effect on search precision by implementing PCA transform [18]. Lastly, the similarity of and is evaluated by the secure Euclidean distance, the minimal distance indicating a higher similarity between these images. We analyze and discuss the relationship between search precision and lowering dimension in Section 7.5.

3. Preliminaries

Prior to discussing the proposed system, the manner of extract the images' vectors and the search accuracy, will be explored.

3.1 Feature Extraction

SURF algorithm is used in our work [21] [22]. This technique is an innovative descriptor and scale and rotation variant detector that matches or often surpasses the patented SIFT algorithm [23] [24] in terms of computation and comparison as well as robustness, distinctiveness and repeatability. As such, this method can enhance the feature extraction efficiency.

The SURF algorithm is used on image *Img* to identify its feature vectors $V = \{v_1, v_2, ..., v_d\}$, where d is interest points number in the given image. Note that diverse images may have varying numbers of descriptors d. PCA transform [18] is then used to reduce the dimensionality of any SURF vectors with ineffectual or low discriminative power. The relationship identified between image searching precision and dimension reduction explored in Section 7.5.

3.2 Mean Average Precision (MPA)

MAP [25] is a technique used to analyze the precision of the request query, more often utilized by well used image search algorithms [17] [18]. The use of MAP allows a mean value quantity to be calculated for a sequence of searches. If for instance *Alice* generates a search query into the Cloud server dataset, a restricted number of outcomes will be generated $\{I_1, I_2, ..., I_{10}\}$. If the two outcomes I_1 and I_{10} are similar to the image in question, we are then able to get the value of *average precision* (AP) in the following manner where $AP = (1/1 + 2/10)/2$ (if S number of comparable images in the database are disregarded in the outcome, then $AP = (1/1 + 2/10 + 0 \times S)/(2 + S)$). The way in which we can calculate MAP in n searches is $= \sum_{i=1}^{n} AP_i / n$.

4. Problem Statement and Security Definition

4.1 Problem Statement

Table 1 lists the most commonly used notations in our work. Assume that the *DO* (*Bob*) has a large-scale image database $DB = \{Img_1, Img_2, ..., Img_n\}$ of *n* sensitive images that he wants to upload to a Cloud server. The SURF local feature vector V_i is then extracted by the *DO* from each image, $Img_i \in DB$, subsequently constructs a proficient searchable tree-based index (*TreeIndex*) by using the *k*-means clustering technique from the extracted features. In light of a security concern, *DO* encrypts the image database *DB* and *Treeindex* before outsourcing them to CS.

The CS is kept from procuring valuable data concerning outsourced data as a result of its encryption, while *DO* is permitted to release such data. The encrypted index tree must encourage an inquiry inside a satisfactory timeframe before restoring those pictures that are most similar those asked for by the smart device client *Alice*.

To look through the remotely stored sensetive image database *DB* with an inquiry *q*, *Alice* produces a safe search request S_q from *q* and after that sends it to CS. The last at that point searches its secure index tree *Treeindex* after receiving the S_q, and the CS subsequently retrieves the candidate list of all available and more similar images. The server then refines the candidate list by completing the Euclidean distance between the feature vectors corresponding to the candidate list and the secure search request (also utilized as a refining vector) of the entered inquery. The *IDs* of top-*t* image are then selected by Cloud server, and the secure and chipertext images that coresponding with these *IDs* are sent back to the smart device client belonging to *Alice*. Subsequently, *Alice* uses his keys to obtain the plaintext images. **Fig. 1** presents the structure of our scheme.

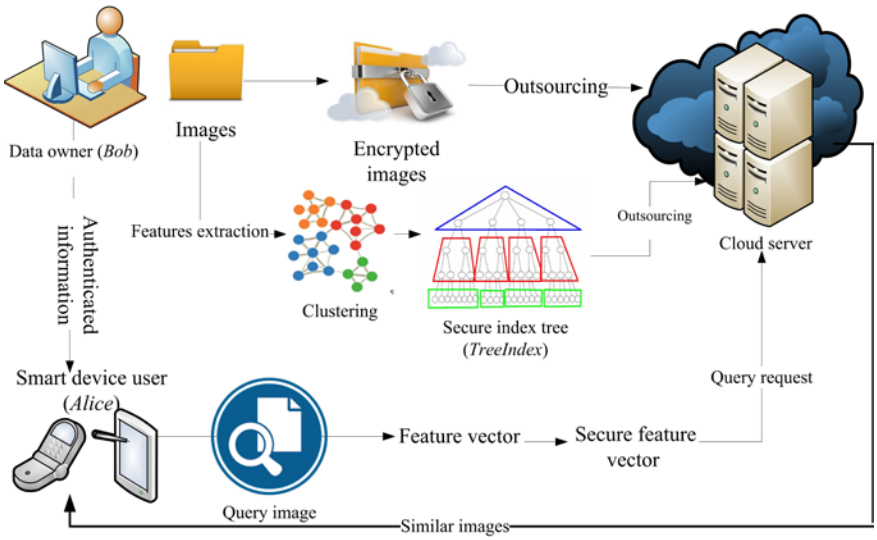

Fig. 1. The proposed scheme architecture

Table 1. Common symbols used

Symbol	Definition
kdata	The symmetric key used for the encryption and decryption of images
DO	Data owner
DB	Extensive image database
M_1, M_2	The encryption keys used the local feature vector of the query and features vector of index tree, respectively
C	The ciphertext of the large-scale image database
q, S_q	Query image and secure query image, respectively
n	Number of sensetive images in *DB* of *DO*
d	Number of descriptors of each image
ID(.)	The image identity in the database

4.2 Security Definition

In order to assist the creation of more realistic applicable outcomes, the definitions of all displayed searchable encryption methods uncover some imperative data to the adversary's server. Such data includes the access patterns as well as and searching patterns (the repetitive sequences of the searches and their corresponding queries). Our work improves security by preventing CS from acquiring the access pattern. We set out what we consider the required improved security local features that should characterize our proposed searchable encryption scheme below.

1. The security of image: No information relating to the stored images should be disclosed by the proposed encryption scheme to the Cloud server with the exception of their image IDs and numbers.
2. Security of search request: *CS* should unable to generate a legitimate request in the absence of a secret key.
3. Security of Index tree: The index tree must not divulge any information pertaining to its substance.
4. Access pattern: The relationship between the index tree and the corresponding image identifiers must not be disclosed to *CS*.

The proposed system employs a symmetric key encryption which can be efficiently searched for sizeable dataset. To keep up with most of the existing searchable encryption schemes, the honest-but-curious server is engaged -upon in the proposed work. The supposition is that CS works in an "hones" way, and that the designed protocol specification is adhered to accurately. In any case, the server will stay "curious" while deriving and evaluating the message-stream that is gathered all through the protocol, thereby acquiring and retaining additional information.

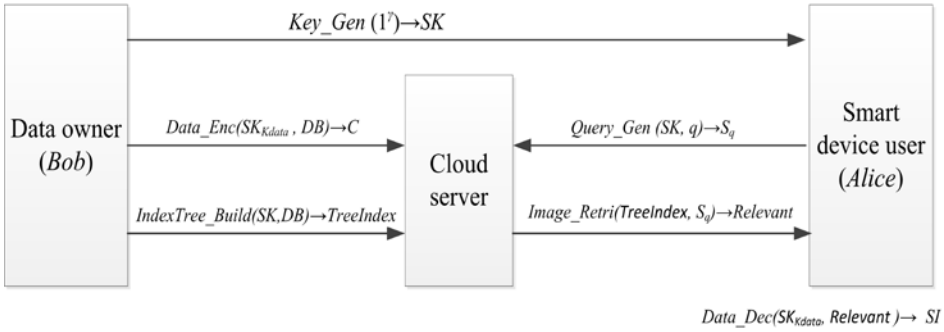

Fig. 2. Technical details of the proposed method

5. The Proposed Scheme Design

This section describes the design of secure encrypted image search framework that performs privacy-preserving and efficiency at the same time. The index-based SE scheme comprise a collection of six polynomial-time algorithms *SE* = (*Key_Gen*, *Data_Enc*, *IndexTree_Build*, *Query_Gen*, *Image_Retri*, *Data_Dec*). The SE process is divided into two stages, in particular, configuration stage and secure image retrieval stage. **Fig. 2** represents the technical detail of the proposed work. In the configuration stage the various notations function as follows:

1. $SK \leftarrow Key\_Gen(1^{\gamma})$: A probabilistic algorithm to generate a key that *DO* executes to start the scheme. The algorithm outputs a secret key $SK \in \{0,1\}^{\gamma}$, where γ represents the security parameter,.
2. $C \leftarrow Data\_Enc(SK_{kdata}, DB)$: The *DO* runs this algorithm, which is a robabilistic algorithm that achieves the encryption of the *DB*. SK_{kdata} is a secret key utilized as an input and in addition an image database $DB = \{Img_1, Img_2, ..., Img_n\}$ and, for the output encrypting dataset, $C = \{c_1, c_2, ..., c_n\}$.
3. $TreeIndex \leftarrow IndexTree\_Build(SK, DB)$: The *DO* runs a deterministic algorithm to create an efficient and secure searchable index tree for searching an image. This algorithm also protects the privacy of the feature vectors that are associated with the nodes of the index tree.

In the secure image retrieval stage the various notations function as follows:

1. $S_q \leftarrow Query\_Gen(SK, q)$: A safe search request S_q is created for a particular image upon the initiation of the smart device client. This deterministic algorithm requires two inputs: the image query *q* and the secret key *SK*, which are both required to make the secret image request S_q.
2. $Relevant \leftarrow Image\_Retri(TreeIndex, S_q)$: A deterministic algorithm that is utilized by the Cloud server to search the index tree *TreeIndex*. Utilizing the secure search image request S_q and the encrypted index tree as inputs, the algorithm outputs *Relevant*, a set of most similar sensetive images that have been encrypted by data owner.
3. $SI \leftarrow Data\_Dec(SK_{kdata}, Relevant)$: A deterministic algorithm that is run by the smart device client to retrieve the plaintext sensetive image. Such algorithm utilizes the

secret key SK_{kdata} as its input and a set of most similar encrypted sensetive images *Relevant* to make an output that comprises a set of similar plaintext sensetive images *SI*.

6. Construction Details

In this part, we clarify the details of our proposed lightweight privacy-preserving CBIR scheme. We will enhance our work building in Subsection 6.3, which further gives a fine-grained user authentication property for searching an image.

6.1 Configuration Stage

Key_Gen: The scheme is initiated by *DO* when calling the $Key\_Gen(1^{\gamma})$ algorithm, which generates the secret key set $SK = \{SK_{kdata}, M_1, M_2\} \xleftarrow{R} \{0,1\}^{\gamma}$. Based on the size *d* of the image descriptor vector, *DO* generates the matrix M_1 and its inverse M_2 in a random fashion, with each matrix having the size (*d* + 2, *d* + 2). The *DO* keeps the secret key that he set himself and the information (SK_{kdata}, M_1) is shared between other authorized data users and *DO* . In this expression, SK_{kdata} denotes the symmetric encryption/decryption key.

Data_Enc: The algorithm $C \leftarrow Data\_Enc(SK_{kdata}, DB)$ is run by *DO* to protect the privacy of the images. *DO* uses pseudo-randomness against the chosen plaintext attack encryption $Enc_{SK_{kdata}}(.)$ to encrypt the collection *DB* before being uploaded into Cloud server, where the secret key is $SK_{kdata} \in SK$. Given the image Img_i and its identifier $ID(Img_i)$, the encrypting image collection *DB* will be $C \leftarrow \{(ID(Img_i), Enc_{SK_{kdata}}(Img_i)\}$ $\forall\, Img_i \in DB$. Image encryption may be achieved by directly using AES, which is modern lightweight ciphers that can treat the images as if they were ordinary data. Specifically, we use AES with CTR modes and counter [26] [27] [28] [29] as an instance of $Enc_{SK_{kdata}}(.)$, and use a 128-bits length secret key as seen in **Fig. 4**. We pad the images file with fake data to conceal their actual size after creating the secure index tree to prevent the addition of random elements within the searchable index.

IndexTree_Build: In order to make the procedure of image searching a speedy and efficient one, it is a necessity for the *DO (Bob)* to create an Index Tree *TreeIndex* for large-scale images database. This is demonstrated in **Fig. 1**. More specifically, we have employed the *k*-means clustering technique during the procedure of classifying the *DB* into a variety of clusters. Such clustering algorithm is one of the least complex technique which utilizes unsupervised learning technique to explain known clustering issues. It works extremely well with vast datasets, while other clustering techniques will in general be more costly. This is because the formula used is dependent on the significance of a pair of images, which can be quantified using the Euclidean distance of their vector features. More specifically the *DO* utilizes the algorithm 1 to divide the larger image datasets into cluters *R*. This continues until all the clusters have fewer than *R* images. The information relayed back from algorithm 1 allows *DO* to have a platform from which it can connect with the descriptor vector feature V_i of every image to a leaf node of the *TreeIndex*.

This platform enables the nodes within comparable clusters to attach to its corresponding non-leaf node. In addition to this, a *k*-dimensional mean descriptor vector VF_{hi} can be designated by the *DO* to its similar non-leaf father node whereby *h* denotes the height of the father node inside the tree index. The *I* denotes the index at *h*. The measure of an element within VF_{hi} is quantified as the mean value for the elements of its connected children. The mean values will have been calculated can then be used throughout the search procedures of the tree to its tree level. VF_{hi} is expressed as follows:

$$VF_{hij} = \sum_{v_i \in Children} v_{ij} \Big/ \left| Children \right| \tag{1}$$

The following subsection introduces a matrix multiplications-based encryption method to protect the confidentiality of the image features vectors that are assigned to the index tree. Afterward, we discuss how such method can be used as a tool for computing the secure Euclidean distance operation in a privacy-preserving mode.

Algorithm 1: Index Tree Building

Input: $\{V_i\}, 1 \le i \le n, R$: ***k***-means cluster number

1. $N_r = n$; //Number of vectors in cluster $r : CL_r$
2. Child_Gen($\{V_i\}_{1 \le i \le n}, R$);
3. Function Child_Gen $(vectors\{V_i\}, R)$ begin
4. *K*-means $(\{V_i\}, R) \rightarrow R$ cluster CL_r;
5. **For** $r = 1$ to R **do**
6. If $N_r > R$ Then
7. Child_Gen $(vectors\{V_i\}_{i \in CL_r}, R)$;
8. **End for**//*r*

Output: $TreeIndex$.

Index Tree Protection: Given that the same encryption process is used in our method for both non-leaf node vectors VF_{hi} and leaf node vectors V_i thus, we only explain the encryption of feature vector V_i for the sake of simplicity. The objective of our work is to encrypt the vectors in a manner which enables it to carry out its Euclidean distance ability without decryption. Before the encryption process, the original representations of these features vectors be augmented to be of *d*+2 dimensions as follows:

$$v'' = (v^T, -0.5 \| v \|^2, 1)^T \tag{2}$$

All of these vectors are then encrypted as follows:

$$v' = E_v(v'', M_2) = v'' M_2^T \tag{3}$$

After the encryption process, *DO* replaces all plaintext feature vectors V_i and VF_{hi} that are allotted to the nodes of *TreeIndex* with their corresponding ciphertexts V_i' and VF_{hi}', respectively. The idea of our method is based on $v.q^T = (vM_2).(M_1 q^T)$, where M_1 and its inverse M_2 represent two encryption keys. Thus the *DO* can store vM_2 instead of *v* at *CS*,

and ensuring that M_2 remains secret from *CS*. *DO* sends $M_1 q^T$ to the Cloud server on all events that they wish to send a query *q*; in this manner the Cloud server can process vq^T without knowing *v* and *q*.

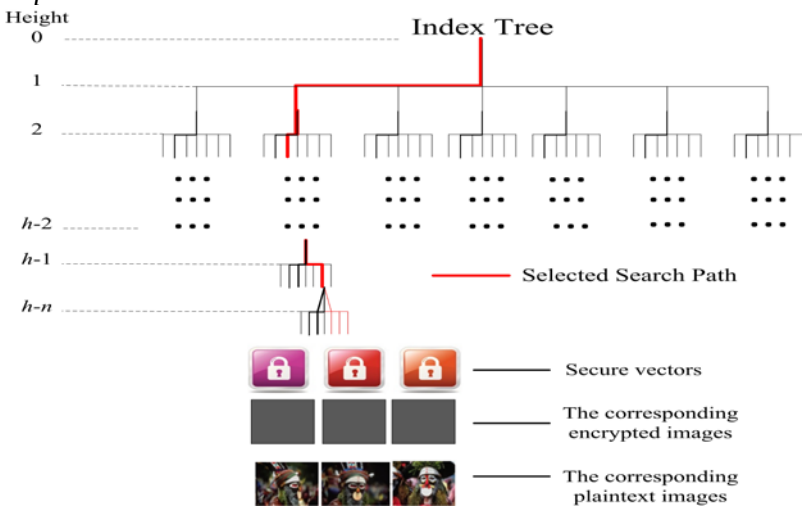

Fig. 3. Image search example

6.2 Secure Image Retrieval Stage

To match her private image, *Alice* uses two algorithms, namely, *Query_Gen* and *Image_Retri*. **Query_Gen**: In this algorithm, to perform a secure search request, *Alice* uses SURF local features to extract the feature vector of search image, and then augments the original representations of the search request vectors with random number $R > 0$, to be of *d*+2 dimensions as follows:

$$q'' = (Rq^T, R, -0.5R \| q \|^2)^T \tag{4}$$

All of these vectors are then encrypted as follows:

$$S_q = q' = E_q(q'', M_1) = M_1 q'' \tag{5}$$

Finally, *Alice* sends secure search request S_q to the Cloud server for image retrieval.

REMARK: the computation in equation (5) does not reveal the key M_1 and *R* of the client to CS. The random number *R* is presented here. *R* expects to keep CS from getting the real distance among *q* and those items from the database. The encryption of feature vector is embraced on only one event time on the *DO* side, this should be noted as it has no impact on the efficiency of the search process.

Image_Retri: After the Cloud server obtains its *Alice* encrypted vectors S_q, as demonstrated in **Fig. 3**, CS will then start from the highest level of the *TreeIndex* and then to the leaf-level when a particular search enquiry is asked for. Consequently, at every level the CS will find the points of entrance to the next level. When moving through every level, by finding the nodes and the vectors VF_{hi} of the lowest Euclidean distance that correspond to each descriptor of

feature vector for image used in search request.

To compare the Euclidean distance of *TreeIndex* node vector V_i or VF_{hi} with that of the search vector q', CS can use their encrypted format without decryption. If we can use $v.q^T$ to represent the distance function $d(v,q) = \sum_{j=1}^{d}(v_j - q_j)^2$, thus it possible for the CS to direct an approximate distance. The following theorem explains the correct distance between the encrypted vectors:

Theorem 1. $(q'.v') = -0.5Rd(v,q)$

Proof. $(q'.v') = (v'')^T.q''\sqrt{a^2+b^2}$

$$= (v^T, -0.5\|v\|^2, 1).(Rq, R, -0.5R\|q\|^2)^T$$
$$= (Rvq - 0.5R\|v\|^2 - 0.5R\|q\|^2)$$
$$= -0.5R(-2vq + \|v\|^2 + \|q\|^2)$$
$$= -0.5R\, d(v,q).$$

Nevertheless, given that $-0.5R$ is a constant, this expression may be removed because of its lack of any impact on the final result. Thusly, $v.q$ can be used by the Cloud server to find the similar match.

The moment at which the CS attains the leaf level (i.e., level *h-n*, please refer **Fig. 3**), it will obtain an index of images that is most relevant with the image requested for during the search. This also means that the Cloud server acquires all of the vectors $\{V_l\}_{1\le l\le M}$ corresponding and relevant with the leaf of *TreeIndex*, so that one congruent list is obtained at the end. This list is known as the candidate list. Consequently, the Cloud server can remove the duplicate images. The following stage includes the refining of the candidate list.

To satisfy such necessities, CS measures the Euclidean distance between the given inquiry request S_q (additionally used as a refining vector) and the vectors in the candidate list. Theorem 1 elucidates how such process is proficient while ensuring security. We apply a procedure to enhance the search accuracy. The CS returns *Relevant* of *t* lower distance and afterward sends these images to the authorized smart device client. The smart device user can then decrypt top-*t* images with his own key SK_{kdata}. Algorithm 2 explains how our protocol retrieves the top-*t* similar images from CS.

To fulfill such necessities, CS measures the Euclidean separation between the given inquiry ask for (additionally used as a refining vector) and the vectors in the competitor list. Hypothesis 1 elucidates how such process is proficient while ensuring security. We apply a refining procedure to enhance the inquiry accuracy. The CS returns comparable encoded pictures Relevant of t bring down separation and afterward sends these pictures to the approved savvy gadget client. The brilliant gadget client would then be able to decode top-t pictures with his very own key . Calculation 2 clarifies our proposed convention for recovering the best t pictures from the remote CS.

Algorithm 2 Image retrieval over encrypted images.

Input: q : query image, $C = c_1,...,c_m$: **C**loud server's large size of encrypted image datasets, *t*: retrieved images number.

Smart Device User Side: {*Alice*}

1. Generate the SURF feature vector q for the query image;
2. Extend and encrypt the local feature vector by (4) and encrypts it by (5), to acquire the secure search request $S_q = q'$;
3. Generate $ESq = E_{xus}(S_q)$, and send (us, ESq) to CS; // *us*: authorized user

Cloud Server Side:

4. $Candidate-list = \theta$;
5. search US_{list} for the user *us*, if found, return *xus*. Otherwise, return $\perp$; // US_{list} : users list
6. use *xus* to decrypt *ESq* to get S_q;

Step1: *{Generating cadidate-list}*

7. Starting the search enquiry S_q from the top level to leaf level of *TreeIndex*, and at every level:
8. Acquire the set of node vectors $\{VF_{hi}\}, 1 \le i \le R$;
9. Calculate the Euclidean distance (as in Theorem 1) between the *Alice*'s query request and the node vector set $\{VF_{hi}\}, 1 \le i \le R$;
10. Finds the entry point of the next level by finding the node with a vector that has the minimal distance to the *Alice*'s search image vector;
11. The process (step 1) repeated until a leaf is reached;
12. Get all vectors $\{V_l\}, 1 \le l \le M$ associated with the indexes of index tree (with leaf level of *TreeIndex*)
13. $Candidate\text{-}list = \{V_l\}, 1 \le l \le M$;

Step2: *{Retrieve similar encrypted images}*

14. Remove the duplicate elements in *Candidate-list*;
15. **For** each $ID(Img_l) \in$ *Candidate-List*
16. Calculate the Euclidean distance (as in Theorem 1) between $V(ID(Img_l))$ and S_q of the received query;
17. **End For**
18. Set similar images vector *SM* to capture the image *IDs* of the minimum *t* distances, $SM = \{sm_1, sm_2,..., sm_t\}$;
19. Set $Relevant = \{C_1 = Retimage(sm_1), C_1 = Retimage(sm_2),..., C_t = Retimage(sm_t)\}$ be the set of the top-*t* similar encrypted images;
20. Send *Relevant* to the end smart device user;

Smart Device User Side: {*Retrieve similar plaintext images*}

21. **For** each encrypted image $C_i, 1 \le i \le t$
22. Generating the plaintext image set $\{SI\}$ by decrypting the encrypted image $C_i \in Relevant$;
23. **End For**

Output: $Relevant$: the collection of top-*t* similar encrypted images.

Data_Dec: Subsequent to getting the set of the top-*t* similar encrypted images, the algorithm $SI \leftarrow DataDec(SK_{kdata}, Relevant)$ is run by the smart device clients with the goal that they can decrypt the encrypted images that correspond to their search requests, consequently acquiring the similar plaintext images *SI*.

6.3 User Authentication

This part of our work will focus on the improvement of our EEIRI system so that it can successfully generate and offer multi-user authentication. In order to support this multi-user approach, the *DO* will provide its secret keys to those smart device users, that have been given permission to do so [30] [31]. However a drawback of this more conventional approach is a high cost. This is because revocation often needs additional round to distribute the new secret keys to the authorized smart device users and requires reconstructing the searchable index tree after every user revocation. This entails the reconstructing of the *Indextree*. It is therefore considered impractical for such dynamic and a large scale operation to be used routinely.
The proposed method is summarised as follow. Let *Aus* represents the authorized smart device clients set. *DO* generatesa secret key *xus* to register each user $us \in Aus$. Afterthat he sends the pairs $[us_j, xus_j](\forall us_j \in Aus)$ to CS, who keeps the received pairs in the user list US_{list}. If a search is performed upon the *q*, the smart device client *us* would initially create a secure request defined as S_q for *q*. $ESq = E_{xus_j}(S_q)$ is calculated, here *E* is a a semantically function that is securely encrypted. Lastly, *us* would sends the search request , which is defined as (us, ESq) to CS.

CS performs a search on the user list US_{list} once the search request is received by the user. The secret key denoted by *xus* will be retrieved in accordance to the smart device user *us*. Output $\perp$ is given in the vent that the search is not found. If the search is found, *xus* will be utilised in order to decrypt *ESq* and to acquire the original S_q. In order to revoke user us_j from *Aus*, Data Owner would send instruction to the Cloud server so that the record $[us_j, xus_j]$ can be removed from US_{list}. The proposed scheme does not need the reconstruction of the searchable index tree, nor the process of distributing new keys. It is presently clear that the users that have been revoked would be unable to perform a search on encrypted images. However, the revocation process does not have an impact upon the authorized smart device users.

7. System Evaluation

7.1 Security Analysis

We posit that the secure searchable index tree, encrypted database, secure query, and secure features vectors will not reveal data to the Cloud server.

7.1.1 Security of Encryption

Clearly, we use AES with CTR mode encryption in our EEIRI to attest that the image collection *DB* will be rendered secure. However, a few issues stay unsolved. For example, the encrypted images may expose data that might be helpful to a Cloud server by righteousness of their size. With the end goal to address this issue, we prescribe enhancing the stored images with padding of further information so their unique sizes are not promptly perceivable. The padding is achieved after creating the protected and secure index tree to prevent the addition of random elements to the searchable index tree. So also, the proposed methodology ensures the protection of access pattern since *CS* stores the encrypted *IDs* of images rather than unsecured plain text.

To build a secure searchable index tree that comprises feature vectors V_i and VF_{hi}, we extend the vectors to (d+2)-dimensions as in equation (2), and then use a matrix multiplications-based encryption method to guarantee the confidentiality of the features vectors as shown in equation (3).

Existing encryption method [32] has proven that if the secret private key is not released to *CS*, it is not possible to obtain the plaintext format of V_i and VF_{hi} from the known cipher text encryption model. In addition, every V_i has an assigned weight that varies between the images. Furthermore, equation 4 and 5 are proved that the query request is protected. With the end goal to evaluate the amount of data leaked to *CS* from the protected from secure vectors V_i, VF_{hi} or secure inquiry request S_q, we measure the dependency between these safe vectors and their relating images *DB*. The reason advising this technique is that low dependency converts into low information leakage. We utilize the mutual information MI entropy [33] to gauge the dependency between two entities. All through our experiments, the protected vectors yield a low MI value, which is equivalent to 0.0017.

7.1.2 Search Unlinkability

A randomization process occurs within the EEIRI design for each search query denoted by *q*. This is conducted through inclusion of (*d+2*)- dimensions random items that are demonstrated in equation (4) as well as encrypted as demonstrated in equation (5). Therefore, there is a difference in terms of encrypted search vectors q' for each search query. This would occur despite having the same search query of image *q* thus ensuring the unlinkability of the various search queries.

7.2 Complexity Analysis

Complexity analysis is performed upon the proposed scheme within framework of the computing time and communication cost. In terms of the complexity of computing time, *n* denotes the number of features vectors of data owner where "(d + 2) - dimensions" occur on each respective vector. At the same time, *Alice's* query, is also a point of "(d + 2)- dimensions" after extension. Transformation and encryption occurs when the index tree is constructed. The dominating process is the encryption process whereby the matrix multiplications based encryption method is applied through amount of times denoted as $log_R(n)$ times and takes the complexity of $(d+2)\times(d+2)$ -matrix with (d +2)-vector at each time. Therefore, the overall computing time complexity of index tree construction is $O((d^2)log_R(n))$ times, whereas that of the query of *Alice* is $O(d^2)$ times.

The computation cost during the secure image retrieval phase needs to run secure Euclidean distance $log_R(n)$ times and takes the complexity of one multiplication of two (d+2)-dimension vectors at each time, thereby the total computing time is $O((d)log_R(n))$ times. Assume that everyelement requires *t* bits, at that point the communication cost for outsourcing the data of *Alice* and *Bob* dominates the factors of $O((td)log_R(n))$ and $O(td)$, respectively.

7.3 Experimental Results

Performance of EEIRI will be analyzed with regards to its efficiency, precision of search, vulnerability of adversary and effectiveness. The experimental results of the presented system have a *DB* originated from familiar Corel dataset with 10,000 color images [34]. We used Matlab R2008a for the experimentation. The experimentation is run with computational specification of 2.5GHz Intel i3-380m CPU with a Windows 7 operating system of 64-bits and RAM 4GB.

During the experiment, *R* of the index tree is set at 50 and each SURF descriptor's size denoted by *m* is set at 4096. Dimension reduction is achieved through the application of PCA transform within the SURF vectors as demonstrated in [18]. The following are values of PCA and *m* for the reduction of the SURF vector dimension; PCA-2048 $(m = 2048)$, PCA-1024 $(m = 1024)$, PCA-512 $(m = 512)$, PCA-256 $(m = 256)$, PCA-128 $(m = 128)$. Normalized vectors are scale by applying a user-specific factor for the conversion to integers which is between 0 and 1. The purpose of this is because the encryption function requires a mandatory integer value.

7.4 Search Efficiency

7.4.1 System Setup

Initially, the SURF technique is applied by *DO* to extract local descriptor feature vectors for search process, with each vector requiring the computation of $O(n)(d)$ -dimensional. Afterward, the *DO* encrypts the features vectors to be used for the search process by utilizing the matrix multiplications-based encryption method. **Fig. 4** demonstrates that the encryption operation cost for a feature vector decreases exponentially to the power of lowering dimension because the size of descriptor *d* is bounded by the principal component analysis PCA-based lowering dimension. After the process of encryption, *DO* constructs the index tree *TreeIndex* with the following operations:

$$\sum_{r=0}^{log_R(n)-1} R^r \; Kmeans(n/R^r, R) \tag{6}$$

Here we analyze the cost of constructing the *TreeIndex* in the context of a reduced dimension and image database size. **Fig. 5** demonstrates that decreasing the dimension can in fact reduce the cost of building a *TreeIndex* because most of the functions employ the *k*-means technique. This means that most of the L_2 distance is calculated with a proportional linear cost to the proportions of the vector. When considering the extent of the database, it is apparent that a larger database size will require a greater cost when constructing the *TreeIndex* (please refer **Fig. 6**). This is due to the fact that *DO* needs supplementary *k*-means clustering on an increased number of vectors.

One-time function is the setup for this system (EEIRI) at the outset and it does not compromise the real time efficiency of the search process. Furthermore, it is possible to optimize the implementation further by modifying to suit the tree construction process, specifically which can process 1 billion 128-dimensional vectors in only 50 minutes by utilizing the *k*-means clustering library [35].

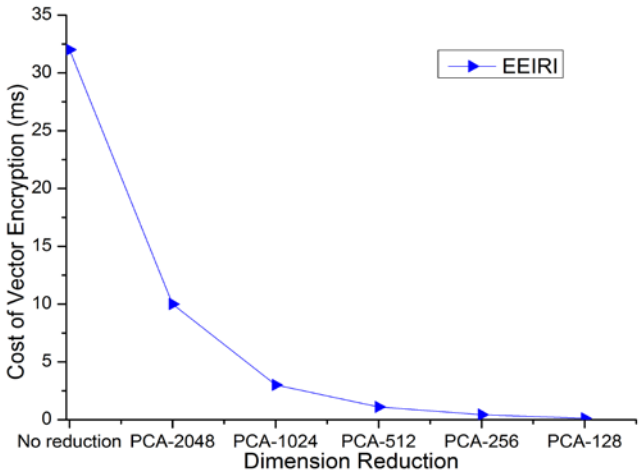

Fig. 4. Cost of vector encryption with various dimension reductions

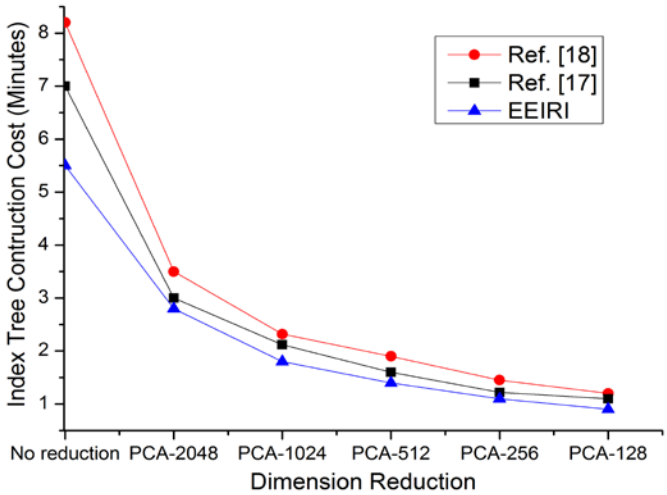

Fig. 5. Cost of index tree construction over 10,000 images of different dimension lowering

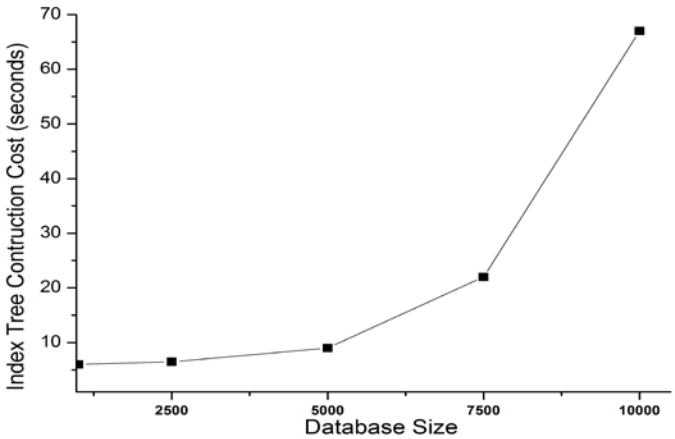

Fig. 6. Cost of index tree construction of different image datasets size over PCA-256 dimension lowering

7.4.2 Search Request Construction

This part of the experiment explores the performance of the search request process of our method, including the construction time and the communication cost. For communication cost, the results of **Fig. 7** are drawn as the communication cost reduction of 10000 queries under various dimension reductions, while **Fig. 8** shows the search request increases using PCA-256 dimension reduction. Our scheme clearly requires a small communication cost. The relatively small yet continual communication cost of the proposed scheme makes it more suitable for use with lightweight devices, including smart devices, when retrieving encrypted images.

Fig. 9 illustrates the search request construction time. To perform a search request construction, Alice generates the secure query request with $O(d^2)$ times of matrix multiplications-based encryption method, which is from 0.012ms to 28ms with variable lowering dimension (see **Fig. 9**). We can see that our method is characterized in which takes a short amount of time to the process of generating search request. This reduced processing time is attributable to the fact that this method does not incorporate an excessive number of time-consuming cryptographic operations, like homomorphic encryption for example.

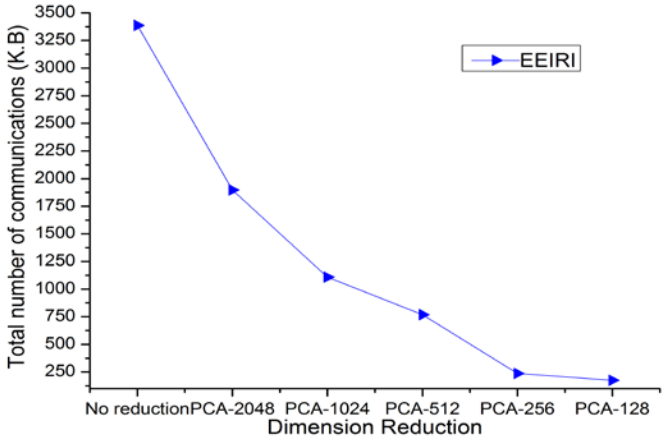

Fig. 7. Communication cost reduction of 10,000 search request with various dimension reductions

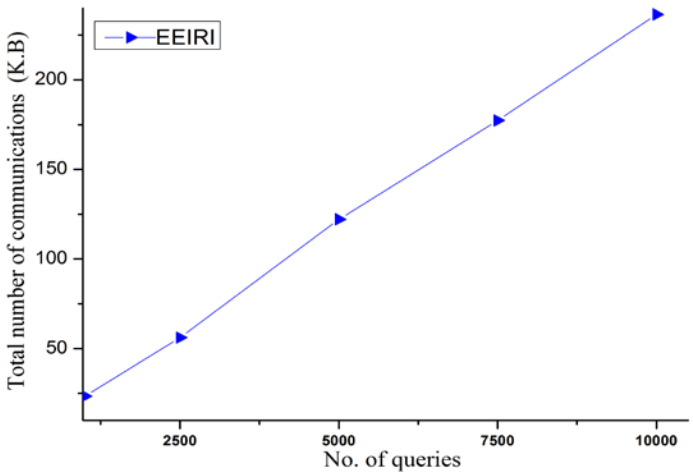

Fig. 8. Communication cost over 10,000 search request using PCA-256 dimension reduction

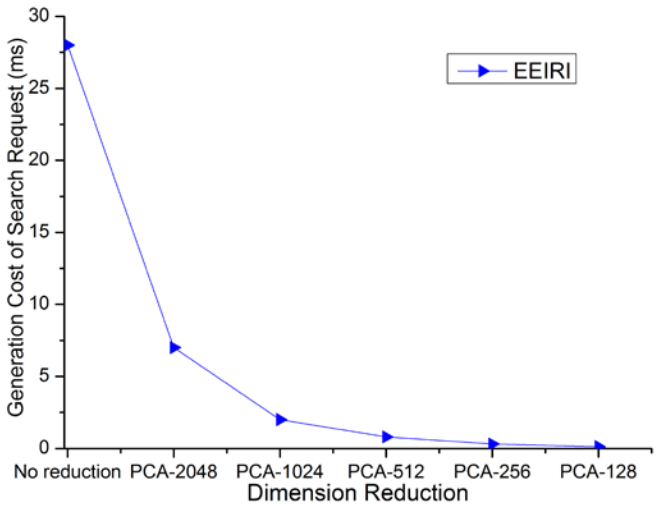

Fig. 9. Query construction time of variable dimension reductions

7.4.3 Search Time

This part proceed to examine the performance of the scheme with specific reference to CS matching time. To perform a search process at the server side, our method requires the searching cost of index tree with $O((d)log_R(n))$ times of the matrix multiplications-based encryption method to compute the search request. As seen in **Fig. 10**, CS searches a database of 10000 images within 148.1ms using EEIRI. Our procedure entails a 7% higher cost of the search when compared with more well-known systems that use plaintext [17] [18]. In addition to this factor, the additional cost of searching time in our method can be considered as a fair cost to achieve the robust and secure matching.

Fig. 11 shows how lowering dimension can further enhance the search efficiency of EEIRI because of the gains in cost among the computations of the matrix multiplications-based encryption method as well as the reduced IO cost from searching *TreeIndex*. More precisely, the PCA-256 can be used to decrease the size of the file from 28 MB to lower than 2.5 MB of 10,000 image vectors which have been designated to the index tree (refer **Fig. 11**). In this manner, the respective vectors can be cached more easily in terms of its memory, which can assist in increasing the efficiency of the searching procedure.

It is apparent that the PCA transformation procedure causes a certain degree of data loss in terms of a particular image features, which may potentially affect the precision of the searches carried out. This may, to a certain extent, offset the association between a more rigorous and efficient system and decrease the descriptor vector that utilizes PCA transformation. The subsequent part of the analysis will discuss in further detail the search precision of the EEIRI as well as the manner in which a right PCA strength is generated helping to generate a more accuracy and successful EEIRI-influenced system.

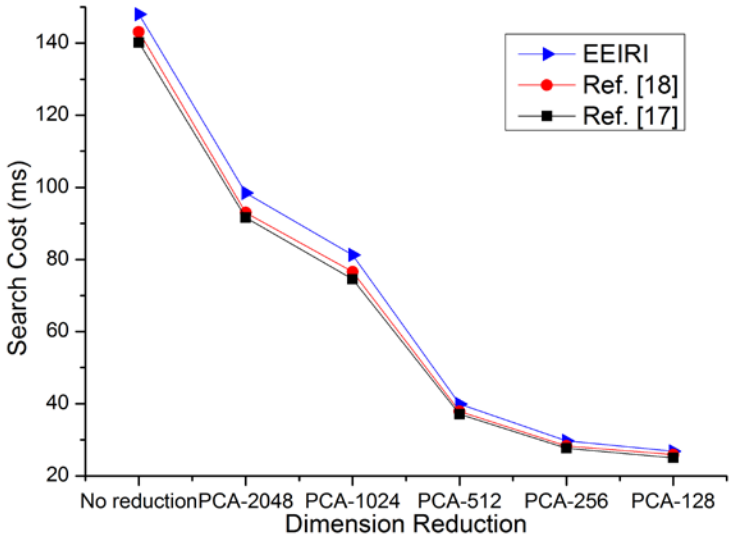

Fig. 10. Search cost of 10,000 images with different dimension reductions

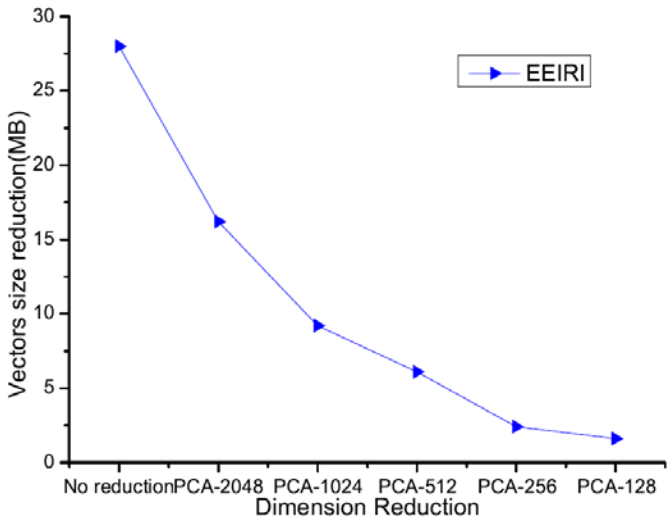

Fig. 11. Lowering of vectors size for 10,000 images with different dimension lowering

7.5 Search Precision

The accuracy with which EEIRI conforms when subject to different dimension reductions is assessed in **Fig. 12**. Our proposed method employs MAP (as introduced in Section 3. 2), and we find that EEIRI is nearly similar to other plaintext image search schemes [17] [18] in term of search precision (only with a 1.7% difference). **Fig. 12** also clearly demonstrates the negative relationship between the dimension reduction over PCA-256and MAP of EEIRI, and the stability of such relationship runs the opposite direction when the dimension reduction is lesser than PCA-256. Therefore, the ideal setting is PCA-256 for balancing the precision and efficiency of the search in practical usage.

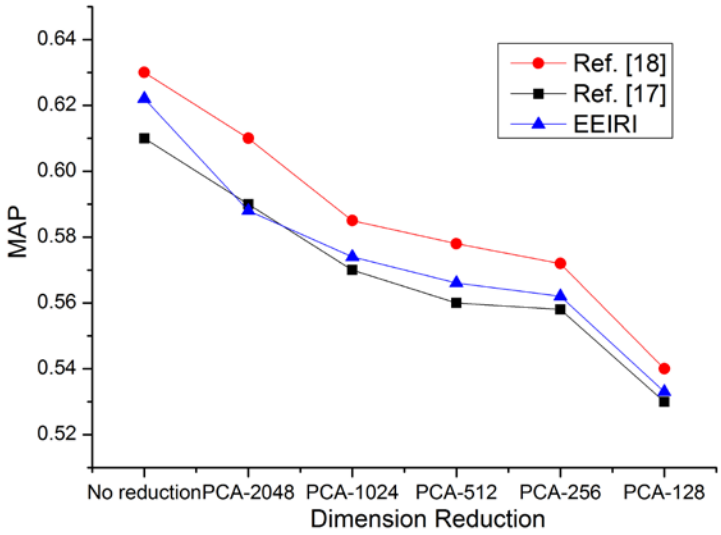

Fig. 12. Affecting search precision over 10,000 images with variable dimension lowering

8. Conclusion

As part of our first endeavor, we attempt to resolve the issue of supporting proficient and content-based image retrieval for smart devices within IoT-Cloud computing systems using encrypted data. Within this context, data privacy must be maintained during the outsourcing process without affecting functionality. In this paper, matrix multiplications and randomization are used to develop a secure process to obtain Euclidean similarity between two feature vector sets. To achieve this, feature vectors are extracted using a SURF descriptor. However, the framework proposed in this paper for secure image matching is not restricted to a specific feature vector.

This work also enhanced the performance when used on large-scale datasets in terms of time efficiency, namely by using common *k*-means clustering techniques to identify each image's representative descriptors as this approach chooses a fewer descriptors. This is used in conjunction with PCA transform which facilitates dimension reduction. Using this approach, it is possible to significantly reduce distance calculation, thus leading to a similar reduction in matching costs. The similarity search is separated into two phases in the proposed scheme. First, a secure search request utilized to set up a candidate list; then the candidate list is refined utilizing a vector intended for this reason likewise recognizes the images with greatest similarity.

We demonstrated the practical value of our work by performing several experiments. We applied our proposed method on over 10000 images, and found that EEIRI only generates minimal losses in terms of search time and precision when compared with other well-known image search algorithms for plaintexts.

Acknowledgement

This work is supported by the Science and Technology Program of Shenzhen of China under Grant Nos. JCYJ20180306124612893, JCYJ20170818160208570 and JCYJ20170307160458368.

References

[1] E. Luo, Q. Liu, J. H. Abawajy, G. Wang, "Privacy-preserving multi-hop profile-matching protocol for proximity mobile social networks," *Future Generation Computer Systems*, vol. 68, pp. 222-233, March, 2017. Article (CrossRef Link)

[2] X. Yang, R. Lu, H. Liang, X. Tang, "SFPM: A secure and fine-grained privacy-preserving matching protocol for mobile social networking," *Big Data Research*, vol. 3, pp. 2-9, April, 2016. Article (CrossRef Link)

[3] S. Wang, J. Y. Mao, N. Archer, "On the performance of B2B e-markets: An analysis of organizational capabilities and market opportunities," *Electronic Commerce Research and Applications*, vol. 11, no. 1, pp. 59-74, February, 2012. Article (CrossRef Link)

[4] J. Y. Chun, D. H. Lee, I. R. Jeong, "Privacy-preserving range set union for rare cases in healthcare data," *IET Communications*, vol. 6, no. 18, pp. 3288-3293, 2012. Article (CrossRef Link)

[5] H. C. Chen, I. You, C. E. Weng, C. H. Cheng, Y. F. Huang, "A security gateway application for End-to-End M2M communications," *Computer Standards and Interfaces*, vol. 44, pp. 85-93, February, 2016. Article (CrossRef Link)

[6] M. Li, S. Yu, Y. Zheng, K. Ren, W. Lou, "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 1, pp. 131-143, March, 2013. Article (CrossRef Link)

[7] D. X. Song, D. Wagner, A. Perrig, "Practical techniques for searches on encrypted data." in *Proc of IEEE Symposium on Security and Privacy*, pp. 44-55, 14-17 May, 2000. Article (CrossRef Link)

[8] D. Boneh, G. Di Crescenzo, R. Ostrovsky, G. Persiano, "Public key encryption with keyword search," in *Proc of Springer International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 506-522, 2-6 May, 2004. Article (CrossRef Link)30

[9] C. Wang, N. Cao, J. Li, K. Ren, W. Lou, "Secure ranked keyword search over encrypted cloud data," in *Proc of IEEE 30th International Conference on Distributed Computing Systems*, pp. 253-262, 21-25 June, 2010. Article (CrossRef Link)

[10] P. Golle, J. Staddon, B. Waters, "Secure conjunctive keyword search over encrypted data," in *Proc of Second International Springer Conference of Applied Cryptography and Network Security*, pp. 31-45, 8-11 June, 2004. Article (CrossRef Link)

[11] W. Sun, B. Wang, N. Cao, M. Li, W. Lou, Y. T. Hou, H. Li, H, "Privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking," in *Proc of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security*, pp. 71-82, 8-10 May, 2013. Article (CrossRef Link)

[12] W. Zhang, Y. Lin, S. Xiao, J. Wu, S. Zhou, S, "Privacy preserving ranked multi-keyword search for multiple data owners in cloud computing," *IEEE Transactions on Computers*, vol. 65, no. 5, pp. 1566-1577, May, 2016. Article (CrossRef Link)

[13] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, W. Lou, "Fuzzy keyword search over encrypted data in cloud computing," in *Proc of INFOCOM*, pp. 1-5, 14-19 March, 2010. Article (CrossRef Link)

[14] C. Liu, L. Zhu, L. Li, Y. Tan, "Fuzzy keyword search on encrypted cloud storage data with small index," in *Proc of IEEE International Conference on cloud Computing and Intelligence Systems*, pp. 269-273, 15-17 Sept., 2011. Article (CrossRef Link)

[15] B. Wang, S. Yu, W. Lou, Y. T. Hou, "Privacy-preserving multi-keyword fuzzy search over encrypted data in the cloud," in *Proc of IEEE Conference on Computer Communications*, pp. 2112-2120, 27 April-2 May, 2014. Article (CrossRef Link)

[16] W. Lu, A. Swaminathan, A.L.Varna, M. Wu, "Enabling search over encrypted multimedia databases," in *Proc. of Springer Media Forensics and Security, SPIE*, vol. 7254, pp. 18-29, 18-22 January, 2009. Article (CrossRef Link)

[17] F. Perronnin, Y. Liu, J. Sanchez, H. Poirier, "Large-scale image retrieval with compressed fisher vectors," in *Proc of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3384-3391, 13-18 June, 2010. Article (CrossRef Link)

[18] M. Douze, A. Ramisa, C. Schmid, "Combining attributes and fisher vectors for efficient image retrieval, " in *Proc of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 745-752, 20-25 June, 2011. Article (CrossRef Link)

[19] C. Y. Hsu, C. S. Lu, S. C. Pei, "Image feature extraction in encrypted domain with privacy-preserving SIFT," *IEEE Transactions on Image Processing*, vol. 21, no. 11, pp. 4593-4607, Nov., 2012. Article (CrossRef Link)

[20] W. Lu, A. L. Varna, M. Wu, "Confidentiality-preserving image search: A comparative study between homomorphic encryption and distance-preserving randomization," *IEEE Access*, vol. 2, pp. 125-141, 2014. Article (CrossRef Link)

[21] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp.91-110, 2004.
Article (CrossRef Link)

[22] S. A. Oliveira, A. R. R. Neto, F. N. Bezerra, "A novel genetic algorithms and surf-based approach for image retargeting," *Expert Systems with Applications*, vol. 44, pp. 332-343, February, 2016.
Article (CrossRef Link)

[23] H. Bay, A. Ess, T. Tuytelaars, L. V. Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no.3, pp. 346-359, June, 2008. Article (CrossRef Link)

[24] D. D. Truong, C. S. Ngoc, V. T. Nguyen, M. T. Tran, A. D. Duong, "Local descriptors without orientation normalization to enhance landmark recognition," in *Proc. of Knowledge and Systems Engineering, Advances in Intelligent Systems and Computing*, vol. 244, pp. 401-413, 2014.
Article (CrossRef Link)

[25] Y. Yue, T. Finley, F. Radlinski, T. Joachims, "A support vector method for optimizing average precision," in *Proc of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 271-278, 23-27 July, 2007. Article (CrossRef Link)

[26] B. Bakhache, J. M. Ghazal, S. E. Assad, "Improvement of the security of zigbee by a new chaotic algorithm," *IEEE Systems Journal*, vol. 8, no. 4, pp.1024-1033, Dec. 2014.
Article (CrossRef Link)

[27] H. Shafagh, A. Hithnawi, A. Droescher, S. Duquennoy, W. Hu, "Talos: Encrypted query processing for the internet of things," in *Proc of the 13th ACM Conference on Embedded Networked Sensor Systems (SenSys'15)*, pp. 197-210, 1-4 November, 2015.
Article (CrossRef Link)

[28] B. Schneier, "Description of a new variable-length key, 64-bit block cipher (blowfish)," in *Fast Software Encryption, Cambridge Security Workshop*, pp. 191-204, 14-16 December, 1994.
Article (CrossRef Link)

[29] P. Kumar, S. B. Rana, "Development of modified {AES} algorithm for data security," *International Journal for Light and Electron Optics*, vol. 127, no. 4, pp. 2341-2345, February, 2016. Article (CrossRef Link)

[30] R. Curtmola, J. Garay, S. Kamara, R. Ostrovsky, "Searchable symmetric encryption: Improved definitions and efficient constructions," in *Proc of the 13th ACM Conference on Computer and Communications Security*, pp. 79-88, October 30 - November 03, 2006. Article (CrossRef Link)

[31] S. Qiu, J. Liu, Y. Shi, M. Li, W. Wang, "Identity-based private matching over outsourced encrypted datasets," *IEEE Transactions on cloud Computing*, vol. 6, no. 3, p. 747-759, December, 2015. Article (CrossRef Link)

[32] W. Du, M. J. Atallah, "Protocols for secure remote database access with approximate matching," *E-Commerce Security and Privacy*, vol. 2, pp. 87-111, 2001. Article (CrossRef Link)

[33] S. Upadhyay, C. Sharma, P. Sharma, P. Bharadwaj, K. Seeja, K "Privacy preserving data mining with 3-d rotation transformation," *Journal of King Saud University - Computer and Information Sciences*, vol. 30, no. 4, pp.524-530, October, 2018.
Article (CrossRef Link)

[34] Corel test set. http://InfoLab.Stanford.EDU/~wangz/image.vary.jpg.tar
[35] W. Liao, "Parallel k-means data clustering", Dec. 5, 2013. http://www.ece.northwestern.edu/~wkliao/Kmeans/index.html

Zaid Ameen Abduljabbar got his bachelor (2002) and master (2006) degrees of computer science from Basrah University, Iraq . Zaid received his PhD in computer engineering in 2017 from Computer Science and Technology, Huazhong University of Science and Technology, China. He is focused on cloud security, searchable encryption systems, similarity measures, Internet of Things, secure computation, biometric, and soft computing. He has published regular papers in many IEEE International Conferences and High-quality papers in SCI journals, and has got the best paper award and published in the 11th International Conference on Green, Pervasive and Cloud Computing (GPC'16), Xi'an, China, 6-8 May 2016. He has always served as a reviewer for several prestigious journals, and has served as a PC chair/PC member of more than 20 international conferences.

Ayad Ibrahim is a doctor in computer science. He was awarded the PhD degree in computer science from HUST University in 2013, his research is about searching the encrypted cloud data. He awarded the master degree in 2005 from Basrah University. The B.E degree was awarded in 2002. His research interests are: Cryptography, Searchable encryption systems, Similarity search, Information retrieval, Record linkage, privacy preserving data mining. He published many a regular scientific papers in the Computer Journal and the IEEE International Conference.

Mohammed Abdulridha Hussain received his Bachelor degree in Computer Engineering in 2004 from University of Basrah, Basrah, Iraq. He received his M.Tech. degree in Computer Science and Engineering in 2009 from GGS Indraprastha University, Delhi, India. Ph.D degree from Huazhong University of Science and Technology, Wuhan, China. His research interests include Network, Network Security, Cloud Security, Web application security and Wireless Sensor Network. He is currently working as a lecturer in Basrah University.

Zaid Alaa Hussien received the B.Sc. degree in Computer Engineering from University of Basrah,Iraq in 2004 , the M.Tech degree in Computer Science and Engineering from Guru Gobind Singh Indraprastha University, India in 2009 and the Ph.D. degree from School of Computer Science and Technology, Huazhong University of Science and Technology, China in 2017. His current research interests include focus on security and integrity issues in cloud computing, big data, and Internet of Things.

Mustafa A. Al Sibahee is an Researcher at Shenzhen Institute of Huazhong University of Science and Technology,Wuhan,China. Lecturer in department of Communication Engineering at Iraq University College, Basrah, Iraq. He is received his Bhd.(2018) and MSc.(2014) degrees in Computer Applied Technology from Huazhong University of science and Technology, Wuhan, China and Bachelor's degree from School of Computer Science at Shatt-alarab University(2010), Basrah, Iraq. His research interests include computer networks, security computer network measurements, machine learning algorithms applications, Wireless Sensor Network (WSN), Software Defined Networking (SDN), embedded systems and Cyber Physical Systems (CPS).

Songfeng Lu is working as a professor in School of Computer Science and Technology, Huazhong University of Science and Technology, China. He received Phd in Computer Science from Huazhong University of Science and Technology in 2001. His research areas include quantum computing, information security and artificial intelligence. You may contact him at lusongfeng@hust.edu.cn.